**SET A**

Python

| def find\_elements\_leq(root, target):  if root is None:  return ""    result = ""    if root.val <= target:  result += str(root.val) + " "    result += find\_elements\_leq(root.left, target)  result += find\_elements\_leq(root.right, target)  return result |
| --- |

Java

| public static String find\_elements\_leq(TreeNode root, int target) {  if (root == null) return "";  String result = "";    if (root.val <= target) {  result += root.val + " ";  }  result += find\_elements\_leq(root.left, target);  result += find\_elements\_leq(root.right, target);  return result;  } |
| --- |

**SET B**

Python

| def find\_elements\_geq(root, target):  if root is None:  return ""    result = ""  if root.val >= target:  result += str(root.val) + " "    result += find\_elements\_geq(root.left, target)  result += find\_elements\_geq(root.right, target)    return result |
| --- |

Java

| public static String find\_elements\_geq(TreeNode root, int target) {  if (root == null) return "";  String result = "";    if (root.val >= target) {  result += root.val + " ";  }    result += find\_elements\_geq(root.left, target);  result += find\_elements\_geq(root.right, target);  return result;  } |
| --- |

**Rubric**

| **Category** | **Marks** |
| --- | --- |
| Proper Method/Function Definition with Proper Parameters | 2 |
| Initiating Proper Resultant Array (pre-order must) | 3 |
| Proper Base Condition with Initialising Result with an Empty List | 3 |
| Proper Recursive Calls | 3 |
| Return Statements | 2 |
| Uses proper syntax, readable naming, and consistent style | 2 |